Chương 2: CSS (Casscading Style Sheets)

Giảng Viên: ThS. Vũ Minh Sang

CSS
Casscading Style Sheets

Nội dung

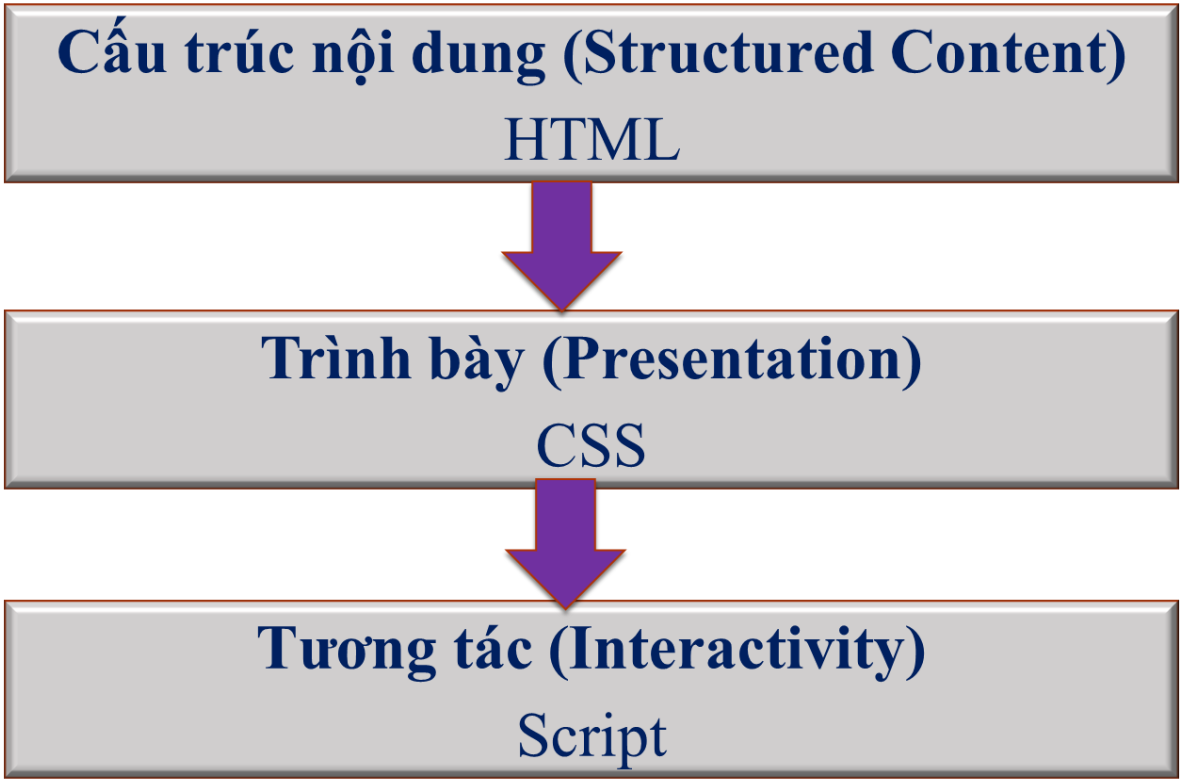
- Giới thiệu CSS
- Thành phần trong CSS
- Selector
- Box model
- Font và text

Giới thiệu CSS

Giới thiệu CSS

- **CSS** = **C**asscading **S**tyle **S**heets
- Dùng để mô tả cách trình bày, hiển thị các thành phần trên trang Web được tạo bằng HTML
- Sử dụng tương tự như dạng TEMPLATE
- Có thể sử dụng lại cho các trang web khác
- Có thể thay đổi thuộc tính từng trang hoặc cả site nhanh chóng (cascading)

Giới thiệu CSS



Giới thiệu CSS

Thành phần trong CSS

Cú pháp

Vùng chọn {

```
        Thuộc tính 1:giá trị 1;
        Thuộc tính 2:giá trị 2;
        ……
        Thuộc tính N:giá trị N;
```

}

- Vùng chọn: tên các **tag**, **id** hoặc **class** của tag
  - `<h1>, <div id="div1">, <h1 class="TieuDe1">`
- Thuộc tính: `width, background-color, position, font`

```
h1{
        font-weight: bold;
        font-size:16pt;
        color: white;
        font-style: italic;
}
```

Ghi chú

- Giống Ghi chú trong C++
- Sử dụng /* Ghichú */
- Ví dụ:

```
SelectorName
{
        Thuộc tính 1:giá trị 1; /*Ghichu1*/
        Thuộc tính 2:giá trị 2; /*Ghichu2*/
        ……..
        Thuộc tính n:giá trị n;
}
```

Phân loại CSS

- Gồm 3 loại:
  - Inline Style Sheet
  - Embedding Style Sheet
  - External Style Sheet

Inline Style Sheet

- Định nghĩa style trong thuộc tính style của từng thẻ HTML
- Cú pháp

```
<tag style="property1: value1;…property N:value N;">… </tag>
```

- Vídụ: `<h1 style="color: yellow">This is yellow </h1>`

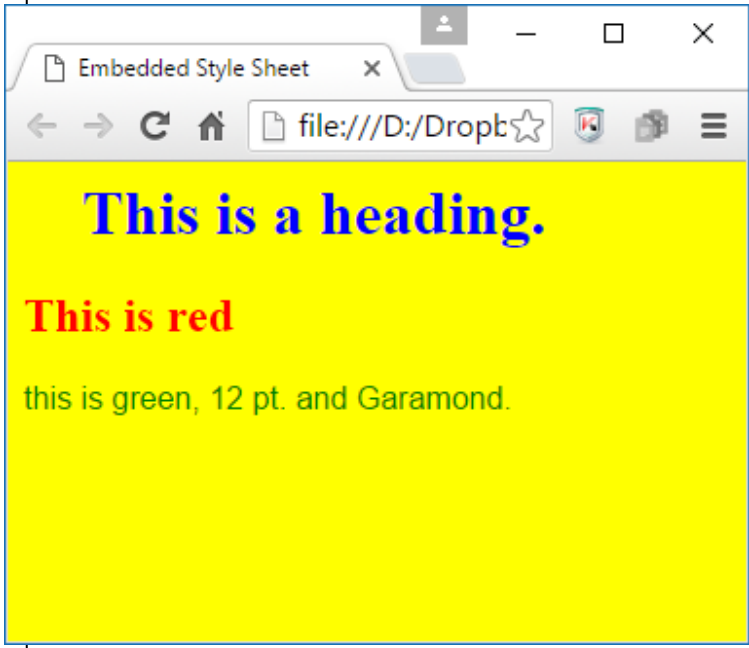
Embedding Style Sheet

- Nhúng thuộc tính css trong thẻ **<style>** của trang HTML
- Cú pháp

```
<head>
<style type="text/css" >
      TagName{
            property 1:value1;
            property 2:value2;
            ……
            property N: valueN;
      }
</style>
</head>
```

Ví dụ

```
<HTML>
<HEAD>
<STYLE TYPE="text/css">
   p{color: green;font-size: 12pt;font-family:
   Arial;}
   h2{color: Red;}
</STYLE>
</HEAD>
<BODY BGCOLOR="#FFFF00">
<h1 style="color:blue;margin-left:30px;">This is
   a heading.</h1>
<h2>This is red</h2>
<p>this is green, 12 pt. and Garamond.</p>
</BODY>
</HTML>
```



External Style Sheet

- Mọi style đều lưu trong file có phần mở rộng là ***.css** (được sử dụng phổ biến)
- Định nghĩa style theo dạng **Embedding Style Sheet**
- Tạo liên kết đến file CSS
  - Liên kết bằng thẻ **<link>.**

```
<head>
   <link rel="stylesheet" href="URL" type="text/css" >
</head>
```

  - Liên kết bằng thẻ **<style>** với @import url.

```
<head>
  <style type="text/css" media="all|print|screen" >
     @import url(URL);
  </style>
</head>
```

Ví dụ External Style Sheet

- Tạo file style.css

```
H2 {
        FONT-WEIGHT: bold;
        FONT-SIZE: 16pt;
        COLOR: white;
        FONT-STYLE: italic;
        FONT-FAMILY: Arial;
        BACKGROUND-COLOR: red;
        font-color: white
}
```

Ví dụ External Style Sheet

- Sử dụng **style.css** trong trang HTML

```
<html>
   <head>
     <title>Cascading Style Sheets</title>
     <link rel="stylesheet" href="style.css" type="text/css" >
   </head>
   <body>
      <h2>This is an H2 </h2>
   </body>
</html>
```

So sánh, đánh giá

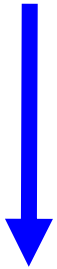
	Inline style sheet	Embedding style sheet	External style sheet
Khai báo	Kiểu 1	Kiểu 2	Kiểu 3

Cú pháp	`<p style="color:red;">`
	`ĐHCNTT`
	`</p>`
		`<style type="text/css"> .tieude1{color=red;}`
		`</style>`
		`<p class="tieude1">`
		`ĐHCNTT`
		`</p>`
			`<link rel="stylesheet" href="style.css">`
			`<p class ="tieude1">`
			`ĐHCNTT`
			`</p>`

Ưu điểm	Dễ quản lý style theo từng tag	+ Dễ quản lý style theo từng tài liệu web + Không cần thêm các trang thông tin khác cho style	+ Thiết lập style cho nhiều tài liệu + Thông tin các style được trình duyệt cache lại
Khuyết điểm	Cần khai báo style trong từng tag của tài liệu	Cần khai báo lại style lại cho các trang khác	+ Tốn thời gian download file .css -> làm chậm quá trình biên dịch web ở trình duyệt trong lần đầu tiên sử dụng

Độ ưu tiên

- Thứ tự độ ưu tiên áp dụng định dạng style dùng trong các trang web (Độ ưu tiên giảm dần)
  - Inline style sheet
  - Embedding style sheet
  - External style sheet
  - Browser Default



Selector

Selector và phạm vi ảnh hưởng

- Là tên 1 style tương ứng với một thành phần được áp định dạng
- Vídụ:

```
.TieuDe1 {
    color: red;
    font-family: Verdana, sans-serif;
}

<h1 class="TieuDe1"> ĐHCNTT</h1>
```

Selector và phạm vi ảnh hưởng (tt)

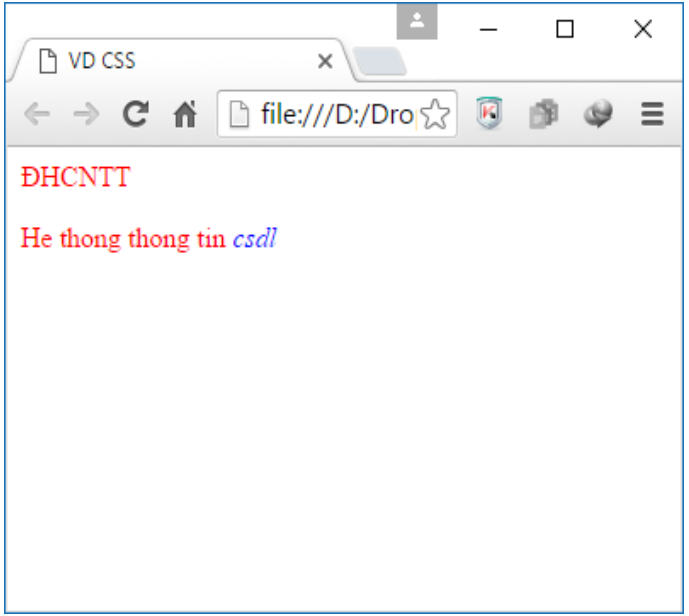
Loại	Mô tả phạm vi ảnh hưởng	Ví dụ
Element	Định dạng áp dụng cho nội dung tất cả các tag **element** trong tài liệu Web	h1{color:red} /*nội dung của thẻ <h1> bị định dạng màu chữ đỏ*/
#id	Định dạng áp dụng cho Nội dung tất cả các **tag** có thuộc tính **id** trong tài liệu Web	#test {color: green;} /* ND của bất kỳ **tag** có thuộc tính **id=test** đều bị định dạng màu chữ=**xanh lá**/

Selector và phạm vi ảnh hưởng (tt)

.class	Định dạng áp dụng cho tất cả các tag có thuộc tính class trong tài liệu Web	.note {color: red;} /* ND của bất kỳ tag có thuộc tính **class=note** đều bị định dạng màu chữ=đỏ */
element.class	Định dạng áp dụng cho nội dung **tag Element** có **thuộc tính class** tương ứng	h1.note { text-decoration: underline;} /*ND của các thẻ <h1> có thuộc tính **class=note** đều bị định dạng gạch chân */

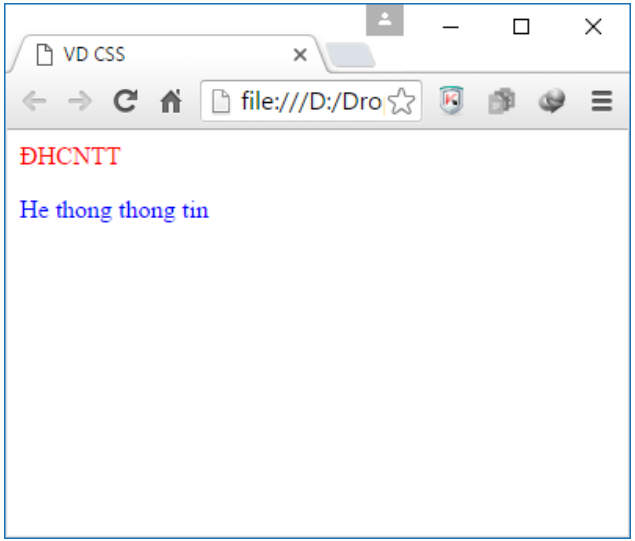
Ví dụ - element

```
<html>
<head>
<title>VD CSS</title>
<style type="text/css">
    p{color:red}
    em{color:blue}
</style>
</head>
<body>
    <p>ĐHCNTT</p>
    <p>He thong thong tin<em>csdl</em></p>
</body>
</html>
```



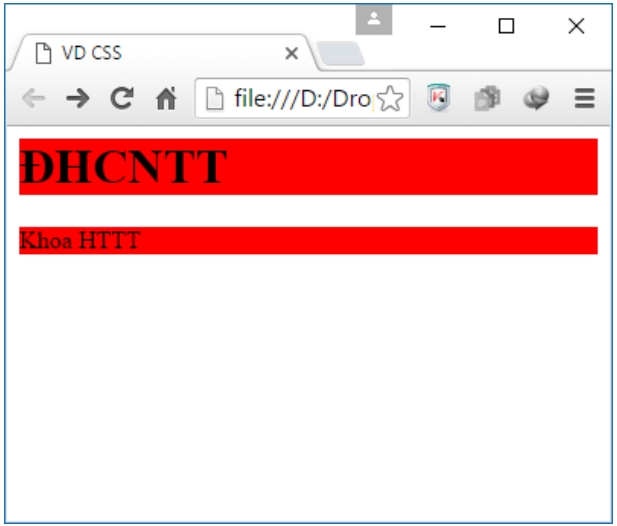
Ví dụ - id

```
<html>
<head>
<title>VD CSS</title>
<style type="text/css">
      #id1{color:red}
      #id2{color:blue}
</style>
</head>
<body>
      <p id="id1">ĐHCNTT</p>
      <p id="id2">He thong thong tin</p>
</body>
</html>
```



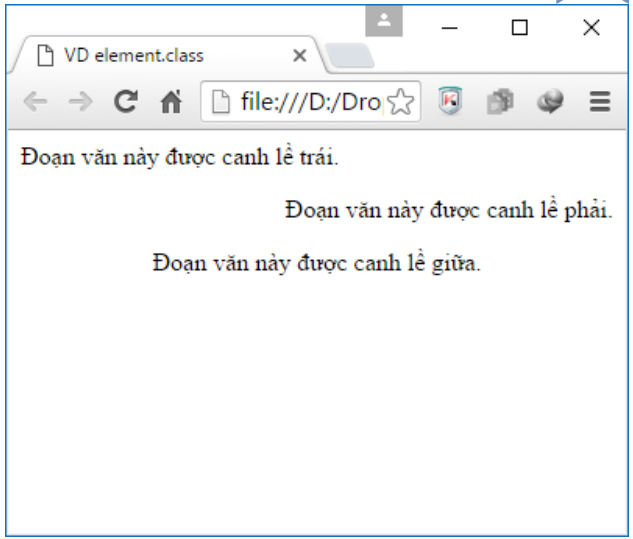
Ví dụ - class

```
<html>
<head>
<title>VD CSS</title>
<style type="text/css">
    .maunen{background-color:red;}
</style>
</head>
<body>
    <h1 class="maunen">ĐHCNTT</h1>
    <p class="maunen">Khoa HTTT</p>
</body>
</html>
```



Ví dụ - element.class

```
<html>
<head>
<title>VD element.class</title>
<style type="text/css">
  p.trai {text-align: left}
  p.phai {text-align: right}
  p.giua {text-align: center}
</style>
</head>
<body>
<p class="trai">Đoạn văn này được canh lề trái.</p>
<p class="phai">Đoạn văn này được canh lề phải.</p>
<p class="giua">Đoạn văn này được canh lề giữa.</p>
</body>
</html>
```



Pseudo-elements

- Pseudo-element: được dùng để chỉ rõ style cho một phần nào đó của phần tử được chọn.
- Ví dụ:
  - Style cho từ đầu tiên, dòng đầu tiên, hoặc là thành phần đầu tiên
  - Thêm nội dung vào trước hoặc sau một thành phần nào đó
- Cú pháp:

```
selector::pseudo-element {
    property:value;
}
```

::first-line Pseudo-elements

- Được dùng để chọn dòng đầu tiên của văn bản
- Ví dụ:

```
p::first-line {
  color: #ff0000;
 font-variant: small-caps;
}
```

- Các thuộc tính có thể dùng:
  - font properties
  - color properties
  - background properties
  - word-spacing
  - letter-spacing
  - text-decoration
  - vertical-align
  - text-transform
  - line-height
  - clear

::first-letter Pseudo-elements

- Được dùng để chọn ký tự đầu tiên trong đoạn văn bản
- Ví dụ:

```
p::first-letter {
  color: #ffff00;
font-size: xx-large;
}
```

- Các thuộc tính có thể dùng:
  - font properties
  - color properties
  - background properties
  - margin properties
  - padding properties
  - border properties
  - text-decoration
  - vertical-align (“float” hoặc “none”)
  - text-transform
  - line-height
  - float
  - clear

::before và ::after Pseudo-elements

- ::before được dùng để chèn nội dung vào phía trước của phần tử
- Ví dụ:

```
h1::before {
    content: url(hinh.gif);
}
```

- ::after được dùng để chèn nội dung vào phía sau của phần tử
- Ví dụ:

```
h1::after {
 content: url(hinh.gif);
}
```

::selection Pseudo-elements

- ::selection sẽ chọn đoạn văn bản được bôi đen bởi người dùng
- Ví dụ:

```
::selection {
    color: red;
    background: yellow;
}
```

- Các thuộc tính:
  - Color
  - Background
  - Cursor
  - Outline.

Pseudo-elements và class

- Ví dụ:

```
p.trai::first-letter {
    color: #ffff00;
    font-size: xx-large;
}
```

Kết hợp nhiều Pseudo-element

- Ví dụ:

```
p::first-letter {
    color: #ffff00;
    font-size: xx-large;
}
p::first-line {
    color: #ff0000;
    font-variant: small-caps;
}
```

Pseudo-classes

- Pseudo-classes: được dùng để chỉ rõ trạng thái đặc biệt của một phần tử được chọn.
- Ví dụ:
  - Style cho phần tử khi rê chuột vào (mouse over)
  - Style cho đường link khi chưa click hoặc (unvisited hoặc visited)
  - Style khi focus
- Cú pháp:

```
selector:pseudo-class {
    property:value;
}
```

Pseudo-classes của thẻ <a>

```
/* unvisited link */
a:link {
    color: #FF0000;
}
/* visited link */
a:visited {
    color: #00FF00;
}
/* mouse over link */
a:hover {
    color: #FF00FF;
}
/* selected link */
a:active {
    color: #0000FF;
}
```

- Lưu ý:
  - a:hover nên đứng ở dưới a:link và a:visited
  - a:active nên đứng dưới a:hover

:first-child Pseudo-classes

- Được dùng định dạng thành phần đầu tiên của nội dung
- Ví dụ: thẻ <p> đầu tiên trong phần nội dung sẽ được định dạng

```
p:first-child {
  color: red;
}
```

- Ví dụ:
  - Tất cả các thẻ <i> đầu tiên của thẻ <p> đều được định dạng

```
p i:first-child {
    color: blue;
}
```

  - Tất cả các thẻ <i> trong thẻ <p> đầu tiên đều được định dạng

```
p:first-child i {
    color: blue;
}
```

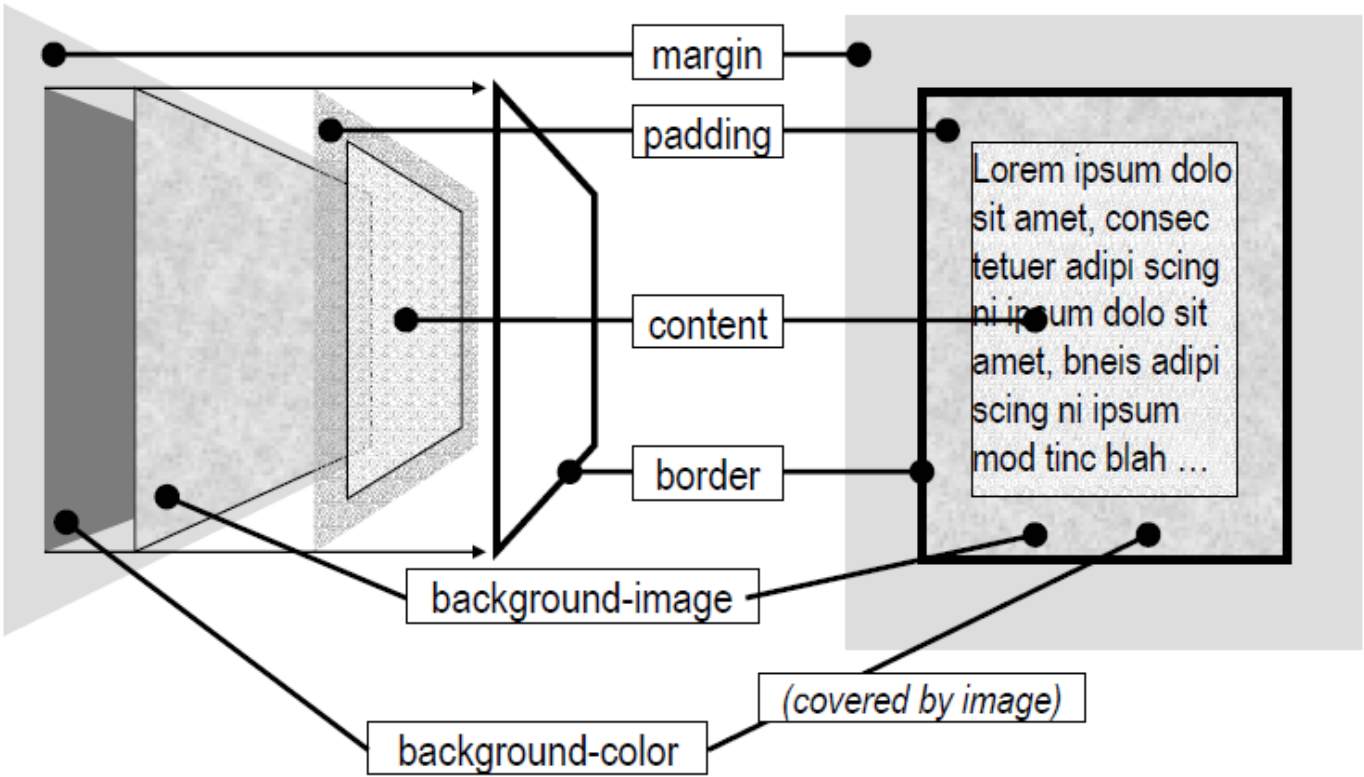
Pseudo-class và class

```
a.highlight:hover {
    color: #ff0000;
}

<p><a class="highlight" href="">Pseudo-classes và classes</a></p>
```

Box model

CSS box model



Background

- Là thành phần nằm sau phần nội dung như là đoạn văn, hình ảnh
- Mở rộng đến phần border, bao gồm cả padding nhưng không vượt quá border.
- Các thuộc tính:
  - **background-color:**[colour-rgb], [colour-hex], [colour-name], transparent
  - **background-image:**[url()],none
    - VD: body { background-image: url(tiles.gif);}
  - **backgroud-position:** top, bottom, left, right, center, [x-% y-%], [x-pos y-pos]
    - VD: background-position: 50% 30px;
  - **background-repeat:** repeat, repeat-x, repeat-y, no-repeat. Dùng để lặp lại hình ảnh theo chiều ngang hoặc chiều dọc

Background

- **background-attachment:** scroll, fixed.
  - VD:

```
div {
    background-image: url(flowers.gif);
    background-attachment: fixed:
}
```

- **background:** background-color background-image backgroundrepeat* background-attachment* backgroundposition*
  - VD:

```
body {
    background: black url(tile.gif) no-repeat top left;
}
```

Border

- Được phân chia với thành phần khác bởi margin
- Các thuộc tính:
  - **border:** border-width border-style border-color
    - VD: p { border: 1px dashed #000; }
  - **border-style:** none, hidden, dotted, dashed, solid, double,groove, ridge, inset, outset
  - **border-color:** [colour-rgb()], [colour-hex], [colour-name]
  - **border-*[top,right,bottom,left]*:** border-width border-style border-color
    - VD: h1 {border-bottom: 1px double green; }

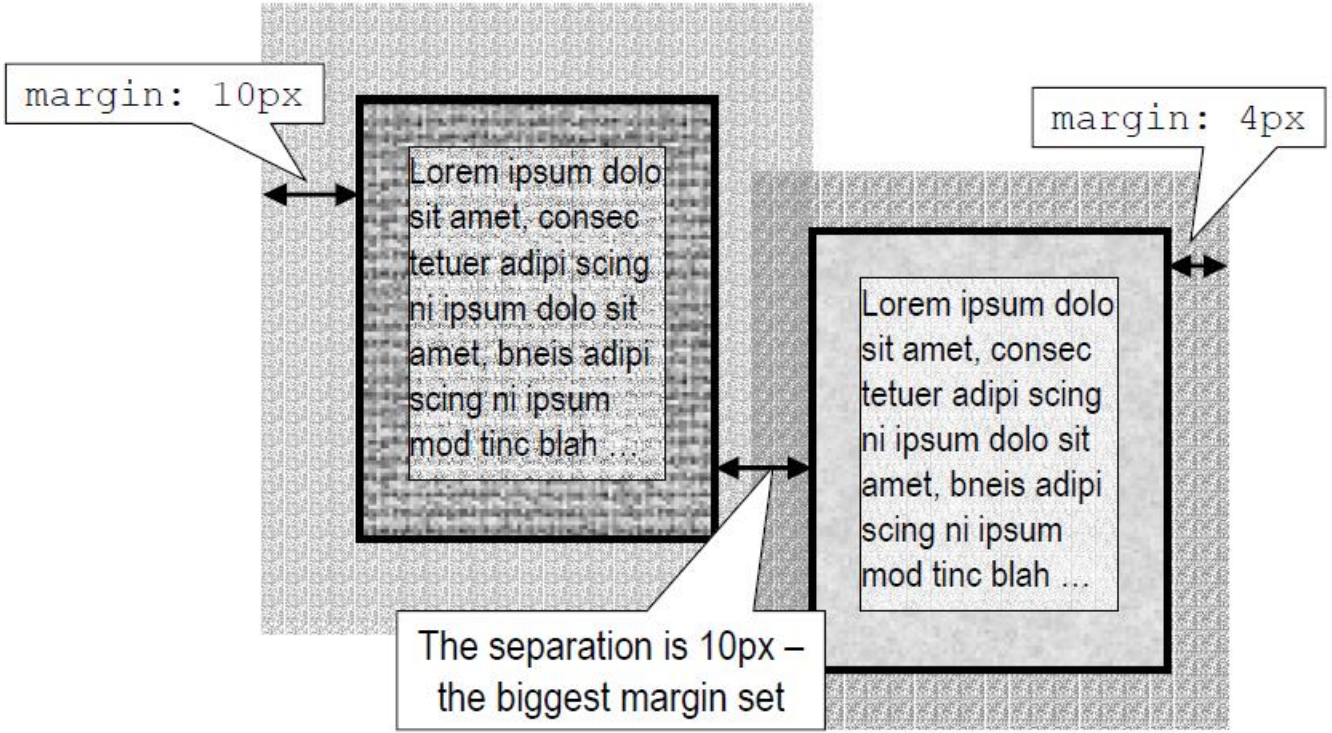
Border

- **border-*[top,right,bottom,left]*-width:** thin, medium, thick, [length]
- **border-*[top,right,bottom,left]*-style:** none, hidden, dotted, dashed, solid, double, groove,ridge, inset, outset
- **border-*[top,right,bottom,left]*-color:** [colour-rgb()], [colour-hex], [colour-name]

Margin

- Cho phép chia các thành phần riêng biệt
- Các thuộc tính:
  - **margin:** margin-top margin-right margin-bottom margin-left. Thiết lập margin cho mỗi thành phần, theo chiều kim đồng hồ bắt đầu từ vị trí top
    - VD: p {margin: 4px 10px 4px 10px; }
  - **margin-*[top,right,bottom,left]*:** auto, [length], [%]
    - VD: li { margin-top: 4em; }
  - **Margin với 1 giá trị:** p {margin: 4px } Tất cả các cạnh của margin sẽ có cùng giá trị (4px)
  - **Margin với 2 giá trị:** p { margin: 10em auto; }
    - Giá trị đầu tiên (10em): cho cạnh trên (top) và cạnh dưới (bottom)
    - Giá trị thứ 2 (auto): cho cạnh trái (left) và cạnh phải (right)

Margin (tt)


margin: 10px
Lorem ipsum dolo sit amet, consec tetuer adipi scing ni ipsum dolo sit amet, bneis adipi scing ni ipsum mod tinc blah …
margin: 4px
Lorem ipsum dolo sit amet, consec tetuer adipi scing ni ipsum dolo sit amet, bneis adipi scing ni ipsum mod tinc blah …
The separation is 10px – the biggest margin set

Padding

- Là khoảng cách giữa nội dung (content) và border
- Các thuộc tính:
  - **padding:**padding-top padding-right padding-bottom padding-left
    - VD: p {padding: 4px 10px 4px 10px; }
  - **Padding với 1 giá trị:** p {padding: 4px } Tất cả các cạnh của pading sẽ có cùng giá trị (4px)
  - **Padding với 2 giá trị:** p {padding: 6px 4px; }
    - Giá trị đầu tiên (6px): cho cạnh trên (top) và cạnh dưới (bottom)
    - Giá trị thứ 2 (4px): cho cạnh trái (left) và cạnh phải (right)
  - **padding-*[top,right,bottom,left]*:**[length], [%]
    - VD: li {padding : 4em; }

Positioning

- Dùng để xác định vị trí của các phần tử
- Vị trí các phần tử được xác định bởi các thuộc tính top, right, bottom, left. Tuy nhiên các thuộc tính này sẽ không hoạt động trừ khi thuộc tính position được thiết lập
- Cú pháp:
  - **position:** static, relative, absolute, fixed
    - **static:** các phần tử được thiết lập mặc định là static và luôn hiển thị bình thường theo thứ tự của trang web. Khi thiết lập static các phần tử ko chịu tác động của các thuộc tính top, right, bottom và left
    - **ralative:** có vị trí tương đối so với vị trí thông thường hiện tại, các phần tử khác sẽ không được đặt bên trái của phần tử này

Positioning

- **fixed**: các phần tử được thiết lập cố định tại vị trí xác định, các thuộc tính top, right, bottom và left được sử dụng cho thuộc tính này.
- **absolute**: vị trí phần tử con được xác định dựa trên phần tử cha của nó.

- **top or bottom**: auto, [%], [length]
- **lefft or right**: auto, [%], [length]
- **overflow**: visible, hidden, scroll, auto: nếu nội dung không vừa với phần tử đang chứa thì có thể thêm thuộc tính overflow để hiện thanh cuộn hoặc không
- **clip:[shape], auto**: dùng để cắt các phần tử theo kích cỡ

Positioning

- **z-index:** auto, [number]: thiết lập thứ tự trước sau của các phần tử

```
#menu {
    position: absolute;
    z-index: 10;
}
```

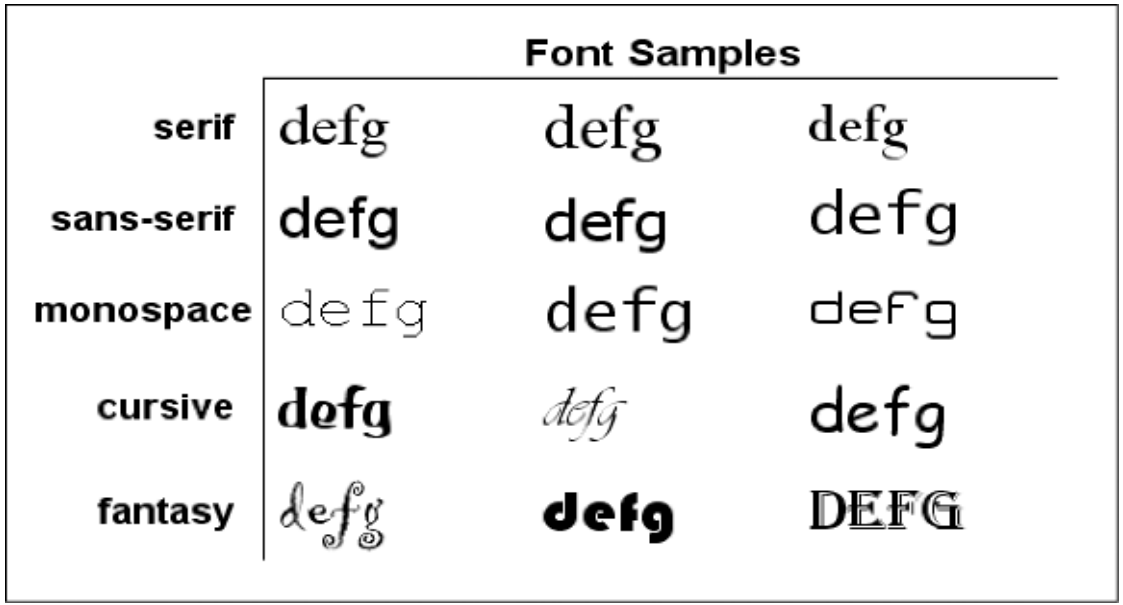
Font và Text

Font và Text

- **generic** và **specific** font
- Thuộc tính
  - `font-family`
  - `font-size, font-style, font-height, font-variant, font-stretch, line-height`
  - `text-align, text-decoration, text-indent, text-shadow, text-transform`
  - `white-space, word-spacing, letter-spacing, direction, unicode-bidi`

Font

- **specific** font là các dạng font như “Times New Roman”, “Arial”
- **generic font** là các dạng font như là “serif”, “san-serif”, “monospace”, “cursive” hoặc “fantasy”.



Font family

- Thuộc tính được dùng để xác định font cho nội dung

```
p {
    font-family: Verdana;
}
```

- Có thể dùng nhiều loại font trong thuộc tính font-family để thay thế khi loại font đấy không có trong máy

```
p {
    font-family: Verdana, Arial, san-serif;
}
```

Thuộc tính của Font

- **font-size:** xx-small, x-small, small, medium, large, x-large, xx-large, smaller, larger, [length], [%]
- **font-style:** normal, italic, oblique
- **font-weight:** normal, bold, bolder, lighter, [100,200, … , 900]
- **font-variant:** normal, small-caps
- **font-stretch:** normal, wider, narrower, ultra-condensed, extra-condensed, condensed, semi-condensed, semi-expanded, expanded, extra-expanded, ultra-expanded
- **line-height:** normal, [number], [length], [%]

Thuộc tính của Font

- Có thể viết cùng lúc nhiều thuộc tính của font trong một dòng

```
font: [style] [variant] [weight] size [/line-height] family
```

  - Những thuộc tính trong **[]** có thể có hoặc không
  - Thuộc tính **size** và **family** bắt buộc phải có
  - Ba thuộc tính đầu có thể dảo thứ tự cho nhau
  - Nếu sử dụng thuộc tính **line-height** thì phải dùng ngay sau thuộc tính **size**

- Ví dụ:

```
p {
    font: italic normal bold 10pt/2em Helvetica, sans-serif;
}
```

Thuộc tính của Text

- **text-align:** left, right, center, justify
- **text-decoration:** none, underline, overline, line-through, blink
- **text-indent:**[length], [%]
- **text-shadow:** none, [color], [length]
- **text-transform:** none, capitalize, uppercase, lowercase
- **vertical-align:** baseline, sub, super, top, text-top, middle, bottom, text-bottom. [length], [%]
- **white-space:** normal, pre, nowrap
- **word-spacing:** normal, [length]
- **letter-spacing:** normal, [length]
- **direction:** ltr, rtl. left to right, right to left
- **unicode-bidi:** normal, embed, bidi-override
- **word-spacing:** value. Khoảng cách giữa các từ
- **letter-spacing:** value. Khoảng cách giữa các ký tự
- **line-height: length.** Khoảng cách giữa các dòng của văn bản

Q & A

Cảm ơn đã theo dõi

Hy vọng cùng nhau đi đến thành công.